

# GDSC Soongsil NFF Study

## 3. Sveltekit Code-jam

GDSC Soongsil 김서현 한재찬

```
const isInterbyOrg = study.lead_organization === interbyorg : true  
const status = filterByStatus ? study.status === filterByStatus : true  
const matchStatus =
```

```
function filterStudies({ studies, filterByOrg = false, filterByStatus = false }) {  
  return studies.filter(study => {  
    const isInterbyOrg = study.lead_organization === interbyorg : true  
    const status = filterByStatus ? study.status === filterByStatus : true  
    const matchStatus =
```



# Rich Harris



# SvelteKit

“Svelte” 를 사용하여  
강력한 성능의 웹 애플리케이션을  
빠르게 개발하기 위한  
프레임워크

# SvelteKit

“Svelte” 는

다른 프레임워크와 달리

런타임 이전인 build 타임에 JS코드로 바뀌어

프레임워크 사용시의 앱 로드 패널티를 가지지 않아도 됨

-> 컴파일러에 가까움

# SvelteKit

- server-side rendering  
(default)
- client-side rendering
- pregeneration

위 모두를 지원

# Svelte..?

## Write less code

HTML, CSS, JS에서  
크게 벗어나지 않는  
친숙한 문법

## No virtual DOM

바닐라 JS로  
virtual DOM을  
대신하는  
효율적으로 컴파일

## Truly reactive

라이브러리없이  
순수 JS로만  
상태 관리



# Component format

Components are the building blocks of Svelte applications. They are written into `.svelte` files, using a superset of HTML.

All three sections – script, styles and markup – are optional.

```
<script>
  // logic goes here
</script>

<!-- markup (zero or more items) goes here -->

<style>
  /* styles go here */
</style>
```



# Nested Component

App.svelte Nested.svelte +

```
1 <script>
2   import Nested from './Nested.svelte';
3 </script>
4
5 <!-- 하위 컴포넌트 -->
6 <Nested/>
```

Result JS output CSS output

This is another paragraph.

App.svelte Nested.svelte x +

```
1 <p>This is another paragraph.</p>
2
```

Result JS output CSS output

This is another paragraph.

# Reactivity

```

App.svelte +
1 <script>
2   let count = 0;
3
4   function incrementCount() {
5     count += 1;
6   }
7 </script>
8
9 <button on:click={incrementCount}>
10   Clicked {count} {count === 1 ? 'time' : 'times'}
11 </button>
12

```

Result	JS output	CSS output
60		
61	function instance(\$\$self, \$\$props, \$\$invalidate) {	
62	let count = 0;	
63		
64	function incrementCount() {	
65	\$\$invalidate(0, count += 1);	
66	}	
67		
68	return [count, incrementCount];	
69	}	

# Reactivity

```
App.svelte +
1 <script>
2   let count = 0;
3   // let doubled = count * 2 => 초반 count = 0의 값만 참고하게 됨
4   $: doubled = count * 2; // [!] count가 바뀔 때마다 갱신됨
5
6   function handleClick() {
7     count += 1;
8   }
9 </script>
```

Result JS output CSS output

Clicked 5 times

5 doubled is 10

**\$:** 코드는 다음과 같이 컴파일됩니다

```
$$self.$$update = () => {
  if ($$self.$$dirty & /*count*/ 1) {
    $: $$invalidate(1, doubled = count * 2);
  }
};
```

# Props

App.svelte | Nested.svelte x +

```
1 <script>
2   export let answer;
3 </script>
4
5 <p>The answer is {answer}</p>
```

Result | JS output | CSS output

The answer is 42

App.svelte | Nested.svelte +

```
1 <script>
2   import Nested from './Nested.svelte';
3 </script>
4
5 <Nested answer={42}/>
```

Result | JS output | CSS output

The answer is 42

# Context API

```
<script>
  import { setContext } from 'svelte';
  const key = Symbol();
  // 첫 번째 인자는 key
  setContext(key, {
    getMap: () => map,
  });
  export { key }
</script>
```

```
<script>
  import { getContext } from 'svelte';
  import { key } from '...';
  const { getMap } = getContext(key);
  const map = getMap();
</script>
```

# Store

```
<script>
  import { setContext } from 'svelte';
  const key = Symbol();
  // 첫 번째 인자는 key
  setContext(key, {
    getMap: () => map,
  });
  export { key }
</script>
```

```
<script>
  import { getContext } from 'svelte';
  import { key } from '...';
  const { getMap } = getContext(key);
  const map = getMap();
</script>
```



# Sveltekit's routing

## Routing

[Edit this page on GitHub](#)

At the heart of SvelteKit is a *filesystem-based router*. The routes of your app – i.e. the URL paths that users can access – are defined by the directories in your codebase:

- `src/routes` is the root route
- `src/routes/about` creates an `/about` route
- `src/routes/blog/[slug]` creates a route with a *parameter*, `slug`, that can be used to load data dynamically when a user requests a page like `/blog/hello-world`

You can change `src/routes` to a different directory by editing the [project config](#).

Each route directory contains one or more *route files*, which can be identified by their `+` prefix.



# +page.svelte

## +page.svelte

A `+page.svelte` component defines a page of your app. By default, pages are rendered both on the server (**SSR**) for the initial request and in the browser (**CSR**) for subsequent navigation.

```
src/routes/+page.svelte
```

```
<h1>Hello and welcome to my site!</h1>
<a href="/about">About my site</a>
```

```
src/routes/about/+page.svelte
```

```
<h1>About this site</h1>
<p>TODO...</p>
<a href="/">Home</a>
```

```
src/routes/blog/[slug]/+page.svelte
```

```
<script lang="ts">
  import type { PageData } from './$types';

  export let data: PageData;
</script>

<h1>{data.title}</h1>
```

# +layout.svelte

## +layout.svelte

To create a layout that applies to every page, make a file called `src/routes/+layout.svelte`. The default layout (the one that SvelteKit uses if you don't bring your own) looks like this...

```
<slot></slot>
```

...but we can add whatever markup, styles and behaviour we want. The only requirement is that the component includes a `<slot>` for the page content. For example, let's add a nav bar:

```
src/routes/+layout.svelte
```

```
<nav>
  <a href="/">Home</a>
  <a href="/about">About</a>
  <a href="/settings">Settings</a>
</nav>

<slot></slot>
```

If we create pages for `/`, `/about` and `/settings` ...

```
src/routes/+page.svelte
```

# Svelte vs React

스벨트와 리액트의 비교		
	스벨트	리액트
개발자	Rich Harris	Jordan Walke
출시일	2016	2011
관리 주체	커뮤니티	Facebook
성능	매우 훌륭함	훌륭함
패키지 크기 (Gzip)	1.6KB	23.2KB (ReactDOM 포함)
반응성	높음	낮음
써드파티 도구	불필요	반드시 필요
개발 속도	매우 빠름	빠름
학습 난이도	매우 쉬움	쉬움
커뮤니티 지원	취약함	훌륭함

출처: Massive Pixel Creation

Next... Code Jam!

npm create svelte@latest  
yarn create svelte  
pnpm create svelte

npm run dev

```
$ npm create svelte@latest svelte-todo
Need to install the following packages:
  create-svelte@latest
Ok to proceed? (y) y
```

```
create-svelte version 2.3.1
```

```
Welcome to SvelteKit!
```

```
✓ Which Svelte app template? » Skeleton project
✓ Add type checking with TypeScript? » Yes, using TypeScript syntax
✓ Add ESLint for code linting? ... No / Yes
✓ Add Prettier for code formatting? ... No / Yes
✓ Add Playwright for browser testing? ... No / Yes
✓ Add Vitest for unit testing? ... No / Yes
```







# Svelte for VS Code

v107.0.2



Svelte



626,406



(36)

Svelte language support for VS Code

Disable



Uninstall



This extension is enabled globally.



Google Developer Student Clubs  
Soongsil University